

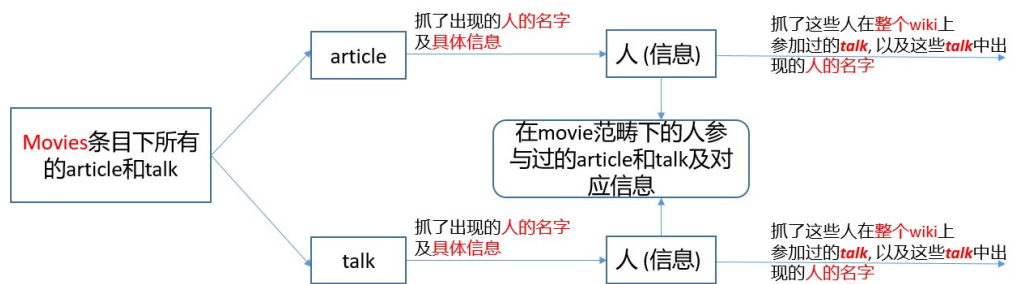
2017 June 25 周报

Junhua Lu

Done

1. 和马老师的简短讨论:
 - a) 快办理手续
 - b) 可能做的话题, 提到是否有用户行为相关的一些东西: (1) 资源调度货物运输行为, 卡车司机的身份信息; (2) 病人的诊疗信息, 时间序列推荐, 类似于 Fan Du 做的一些东西. 在那边还有一些有意思的数据集或者问题, 后面可以再定.
2. 写了大量代码, 做的数据抽取, 以及别人之前处理过程出现的各种问题 | 不是电影的算成电影, 时间戳的格式不对, category 不对等等.

我们期望的是, 在这些数据基本问题解决后, 首先对电影和对应的 talk 匹配, 去除 talk 为空的所有电影及对应 talk, 去除 template. 做一些统计, 诸如电影类别分布, 电影时间跨度分布, 人员分组分布. 然后, 按照之前过滤规则, 消除 ip 地址用户和机器人用户, 在所有可用电影里构建网络, 再进行后面模型运行.



3. 开始学习 Node.js 等相关的后端知识, 结合之前投稿月别人写的一些代码.

To Do

1. wiki 数据处理完, 选好应该要做的数据部分; 模型输入整理完
2. 专利修改, 今年投稿论文视图的修改
3. NodeJS 的继续学习
4. 回家办一些手续

论文阅读:

EuroVis 17 *Graffinity: Visualizing Connectivity in Large Graphs* 文章讲的是图中 connectivity 的东西, 有点类似于 Egolines 里面做的东西, 但是从另外一个角度来提供聚合信息和详细信息. 文章结构清晰, 配图明了, 可能与通讯作者 M. Meyer 有点关系. 系统也是十分简洁, 与前面几位作者的作品看起来风格类似, 清新实用, 没有用特别复杂的算法来数据关系, 而是从问题出发, 抽象到一定层面用可视化方法来解决. 这样还能避免 scalability 的问题.

CHI 17 *User-Guided Synthesis of Interactive Diagrams* 一开始以为是可视化的类似于半自动生成交互图形的文章, 其实偏向于 diagram 制作, 或者说把可视化一些技巧应用到这种 diagram 中来. 这种 diagram, 举个例子就是物理课上弹簧受力模型的动态图, 文章以此做 demo, 赋以整个图形编辑、操作系统以 layout 约束与交互约束, 并在适时提供用户交互的点. 这个作品能大大减少制作成本与时间, 并且不一定需要有编程背景的人才能做.

CSCW 17 Best Paper (Computer Supported Cooperative Work and Social Computing) *Anyone Can Become a Troll: Causes of Trolling Behavior in Online Discussions* 互联网中(社交、评论的)往往会有一些用户专门来引战, 或者说搞事情. 别人越是骂他他越开心, 对于这种行为, 在网络上被称为 trolling, 这种行为就和发帖机灌水一样对网络环境影响不好. 文章通过两种实验: 1 模拟聊天室实验 2 网上爬的 CNN 评论数据集, 来获得一些结论, 并且这些结论不同于传统的一些研究的结果. 其中也是采用了简单的逻辑回归.

WWW2017 Best Paper Honorable Mention *An Army of Me: Sockpuppets in Online Discussion Communities* 同样是上面作者的一个研究, 可以说这些问题很常见但是作为研究都是第一次见: 本文分析的是网络上的马甲行为. 作者在这里定义的马甲比传统研究的定义(马甲是欺骗为主的定义)更广一点, 讲我们所说的马甲那种引导舆论的功能也考虑在内, 从一个 data driven 角度出发, 用了多个数据集, 发现一些特点、验证一些因素; 并用这些因素作为预测的自变量去预测, 取得了可观的效果. 值得注意的是数据来源都比较简单(从数据爬取角度), 但获得 ground truth 仍然需要一些专家经验. 一方面这个问题可以拿来练; 另一方面可视化是不是能更好的辅助这个探测过程、亦或是对结果更好的解读提供上下文, 是可以考虑的.

EuroVis 17 *CoreFlow: Extracting and Visualizing Branching Patterns from Event Sequences* 对于事件序列, 之前也有各种诸如时序的 frequent pattern mining 及可视化. 但是本文出发点别出心裁, 他是抓 branching pattern, 打个比方就是把 sequence 看做旅行者路线, 本文的算法结合可视化展示了路线里重要 milestone 的 overview. 同样又是转换了一个视角, 正如上面第一篇不注重于点线而是连接性一样, 感觉很有启发性.